International Journal of Science Engineering and Advance Technology, IJSEAT, Vol. 5, Issue 1
ISSN 2321-6905
January -2017

International Journal of Science Engineering and Advance Technology

Recursive Approach to the Design of a Parallel Self Timed Adder

Kakidi Kadavath Beeran , R. Prasad
M.Tech Student , Assistant Professor
Department of ECE,
Sri Aditya Engineering College (JNTUK), Surampalem, Andhra Pradesh - 533 437
kkbeeran414@gmail.com, prasadrayi@gmail.com

ABSTRACT - As innovation scales down into the lower nanometer values control, postpone region and recurrence gets to be im¬portant parameters for the examination and plan of any circuits. This short exhibits a parallel single-rail self-coordinated viper. It depends on a recursive definition for performing multi bit double expansion. The operation is parallel for those bits that needn't bother with any convey chain spread. Therefore, the outline achieves logarithmic performance over arbitrary operand conditions with no extraordinary speedup hardware or look-ahead pattern. A viable execution is furnished alongside a finish recognition unit. The usage is regular and does not have any commonsense confinements of high fanouts. A high fan-in entryway is required however yet this is unavoidable for offbeat rationale and is overseen by associating the transistors in parallel. Reproductions have been performed utilizing an industry standard toolbox confirm the reasonableness and prevalence of the proposed approach over existing offbeat adders.

Keywords: - Digital arithmetic, Binary adders, Recursive adder.

I. INTRODUCTION

Double expansion is the absolute most imperative operation that a processor performs. The majority of the adders have been intended for synchronous circuits despite the fact that there is a solid enthusiasm for clock less circuits [1].Asynchronous circuits don't expect any quantization of time. Along these lines, they hold extraordinary potential for rationale outline as they are free from a few issues of timed (synchronous) circuits. On a fundamental level, rationale stream in offbeat circuits is controlled by On the other hand wave pipelining (or max-imal rate pipelining) is a strategy that can apply pipelined contributions before the yields are balanced out [7]. The proposed circuit oversees programmed single-rail pipelining of the convey inputs isolated by engendering and inertial postponements of the entryways in the circuit way.

II. SELF-TIMED ADDERS

Self-planned alludes to rationale circuits that rely on upon timing suppositions for the right operation. Self-coordinated adders can possibly run speedier arrived at the midpoint of for element information, as early fulfillment detecting can maintain a strategic distance from the requirement for the most pessimistic scenario packaged defer component of synchronous circuits.

A. PIPELINED ADDERS USING SINGLE-RAIL

Information Encoding

The offbeat Req/Ack handshake can be utilized to empower the snake obstruct and also to set up the stream of convey signs. In the vast majority of the cases, a double rail convey tradition is utilized for inside bitwise stream of convey yields. These double rail signs can speak to more than two rationale values (invalid, 0, 1), and along these lines can be utilized to create bit-level affirmation when a bit operation is finished. Last finish is detected when all piece Ack signs are gotten (high). The convey fruition detecting viper is a case of a pipelined snake [8], which utilizes full viper (FA) useful pieces adjusted for double rail convey. Then again, a theoretical fruition viper is proposed in [9]. It utilizes purported prematurely end rationale and early fruition to choose the correct culmination reaction from various settled postpone lines. In any case, the prematurely end rationale usage is costly because of high fan-in necessities.

B. DELAY INSENSITIVE ADDERS USING DUAL -

Rail Encoding

Defer uncaring adders are offbeat adders that state packaging limitations or DI operations. In this manner, they can effectively work in nearness of limited yet obscure entryway and wire postpones [2].There are numerous variations of DI adders, for example, DI swell convey snake (DIRCA) and DI convey look-ahead viper (DICLA). DI adders utilize double rail encoding and are expected to expand intricacy. In spite of the fact that double rail encoding duplicates the wire many-sided quality, they can in any case be utilized to deliver circuits almost as productive as that of the single-rail variations utilizing dynamic rationale or nMOS just outlines. An illustration 40 transistors for each piece DIRCA viper is exhibited in [8] while the traditional CMOS RCA utilizes 28 transistors. Like CLA, the DICLA characterizes convey proliferate, create, and murder conditions as far as double rail encoding [8]. They don't associate the convey motions in a chain but instead arrange them in a progressive tree. Along these lines, they can possibly work speedier when there is long convey chain. A further enhancement is given from the perception that double rail encoding rationale can profit by settling of either the 0 or 1 way. Double rail rationale require not sit tight for both ways to be assessed. In this manner, it is conceivable to further accelerate the convey look-ahead hardware to send convey create/convey execute signs to any level in the tree. This is explained in [8] and alluded as DICLA with speedup hardware (DICLASP).

III. PARALLEL SELF TIMED ADDERS

In this area, the engineering and hypothesis behind PASTA is displayed. The viper first acknowledges two information operands to perform half increases for every piece. Along these lines, it emphasizes utilizing prior created convey and totals to perform half-increases more than once until all convey bits are devoured and settled at zero level.

A. Design of PASTA

The general design of the viper is appeared in Fig.1. The determination contribution for two-input multiplexers relates to the Req handshake flag and will be a solitary 0 to 1 move indicated by SEL. It will at first select the genuine operands amid SEL = 0 and will change to input/convey ways for consequent emphasess utilizing SEL = 1. The input way from the HAs empowers the different cycles to proceed until the finishing when all convey signs will accept zero qualities.

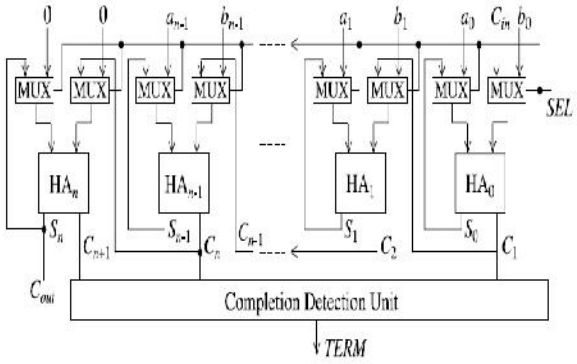


Fig.1 Block diagram of PASTA.

B. State Diagrams

In Fig. 2, two state diagrams are drawn for the initial phase and the iterative phase of the proposed architecture. Each state is represented by (C_{i+1} S_i) pair where C_{i+1}, S_i represents carry out and sum values, respectively, from the i^{th} bit adder block. During the initial phase, the circuit merely works as a combinational HA operating in fundamental mode. It is apparent that due to the use of HAs instead of FAs, state *(*11*)* cannot appear. During the iterative phase (SEL = 1), the feedback path through multiplexer block is activated. The carry transitions (C_i) are allowed as many times as needed to complete the recursion. From the definition of fundamental mode circuits, the present design cannot be considered as a fundamental mode circuit as the input–outputs will go through several transitions before producing the final output. It is not a Muller circuit working outside the fundamental mode either as internally; several transitions will take place, as shown in the state diagram. This is analogous to cyclic sequential circuits where gate delays are utilized to separate individual states [4].

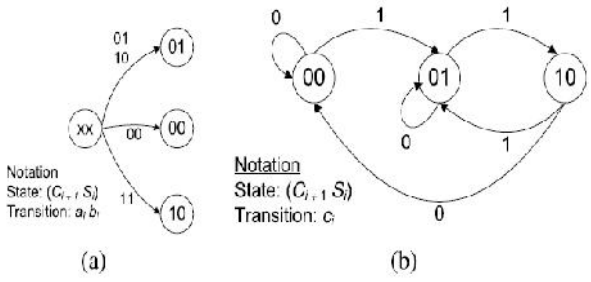


Fig.2. State diagrams for PASTA.
(a) Initial phase. (b) Iterative phase.

C. RECURSIVE FORMAT FOR BINARY ADDITION

Let S^j_i, and C^j_{i+1} denote the sum and carry, respectively, for i^{th} bit at the j^{th} iteration. The initial condition (j = 0) for addition is formulated as follows:

$$S^0_i = a_i \oplus b_i$$
$$C^0_{i+1} = a_i . b_i \quad (1)$$

The j th iteration for the recursive addition is formulated by

$$S^j_i = S^j_j \oplus C^{j-1}_i \quad 0<i<n \quad (2)$$
$$C^j_{i+1} = S^j_j . C^{j-1}_i \quad 0<i<n \quad (3)$$

The recursion is terminated at k^{th} iteration when the following condition is met:

$$C_n{}^k + C_{n-1}{}^k + \ldots\ldots + C_1{}^k = 0 \quad 0<k<n$$

Now, the correctness of the recursive formulation is inductively proved as follows.

Theorem 1: The recursive formulation of (1)–(4) will produce correct sum for any number of bits and will terminate within a finite time.

Proof: We prove the correctness of the algorithm by induction on the required number of iterations for completing the addition (meeting the terminating condition).

Basis: Consider the operand choices for which no carry propagation is required, i.e., $C^0_j = 0$ for $\forall i$, $i \in [0...n]$. The proposed formulation will produce the correct result by a single-bit computation time and terminate instantly as (4) is met.

Induction: Assume that $C^k_{i+1} != 0$ for some i^{th} bit at k^{th} iteration. Let l be such a bit for which $C^k_{l+1} = 1$. We show that it will be successfully transmitted to next higher bit in the $(k + 1)^{th}$ iteration.

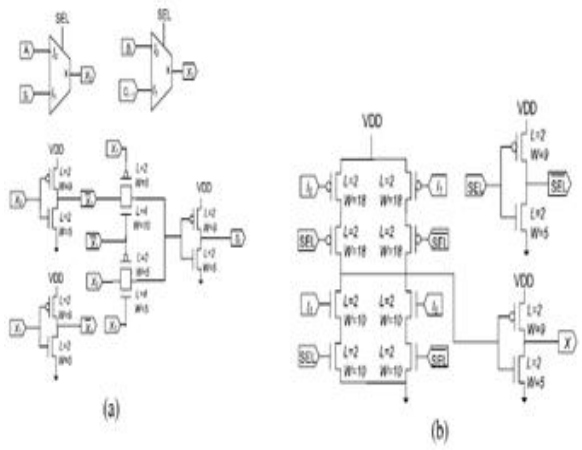
As shown in the state diagram, the k^{th} iteration of l^{th} bit state (C^k_{l+1}, S^k_l) and $(l + 1)^{th}$ bit state (C^k_{l+2}, S^k_{l+1}) could be in any of (0, 0), (0, 1), or (1, 0) states. As $C^k_{l+1} = 1$, it implies that $S^k_l = 0$. Hence, from (3), $C^{k+1}_{l+1} = 0$ for any input condition between0 to l bits.

We now consider the $(l + 1)^{th}$ bit state (C^k_{l+2}, S^k_{l+1}) for k^{th} iteration. It could also be in any of (0, 0), (0, 1), or (1, 0) states. In $(k+1)^{th}$ iteration, the (0, 0) and (1, 0) states from the k^{th} iteration will correctly produce output of (0, 1) following (2) and (3). For(0, 1) state, the carry successfully propagates through this bit level following (3).Thus, all the single-bit adders will successfully kill or propagate the carries until all carries are zero fulfilling the terminating condition.

The mathematical form presented above is valid under the condition that the iterations progress synchronously for all bit levels and the required input and outputs for a specific iteration will also being synchrony with the progress of one iteration. In the next section, we present an implementation of the proposed architecture which is subsequently verified using simulations.

IV. Design of PASTA

A CMOS implementation for the recursive circuit is shown in Fig.3. For multiplexers and AND gates we have used Xilinx ISE implementations while for the XOR gate we have used the faster ten transistor implementation based on transmission gate XOR to match the delay with AND gates [4]. The completion detection following (4) is negated to obtain an active high completion signal (TERM). This requires a large fan-in *n*-input NOR gate. Therefore, an alternative more practical pseudo-nMOS ratioed design is used. The resulting design is shown in Fig. 3(d). Using the pseudo-nMOS design, the completion unit avoids the high fan-in problem as all the connections are parallel. The pMOS transistor connected to *V*DD of these ratio-ed design acts as a load register, resulting in static current drain when some of the nMOS transistors are on simultaneously. In addition to the *Ci* s, the negative of SEL signal is also included for the TERM signal to ensure that the completion cannot be accidentally turned on during the initial selection phase of the actual inputs. It also prevents the pMOS pull up transistor from being always on. Hence, static current will only be flowing for the duration of the actual computation.



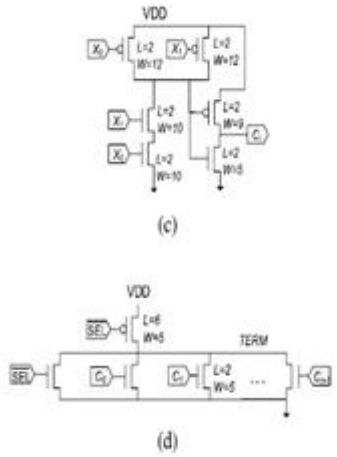


Fig.3.CMOS implementation of PASTA. (a) Single-bit sum module.
(b) 2×1 MUX for the 1 bit adder.
(c) Single-bit carry module.
(d) Completion signal detection circuit.

V. SIMULATION RESULTS

The corresponding simulation results of the PASTA adders are shown below. All the synthesis and simulation results are performed using Verilog HDL. The synthesis and simulation are performed on Xilinx ISE 14.4. The simulation results are shown below figures.

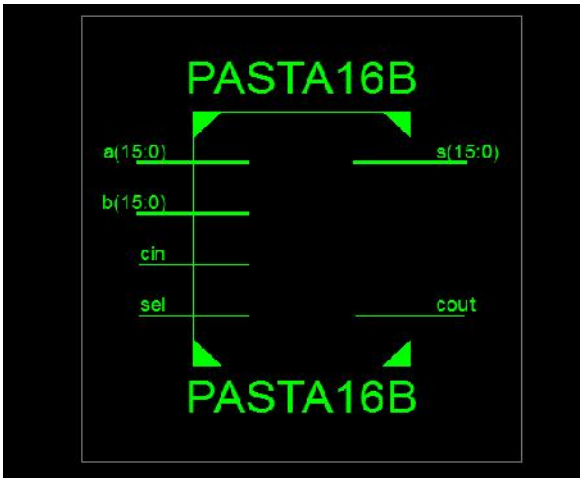


Figure 4: RTL schematic of Top-level 16 bit PASTA adders

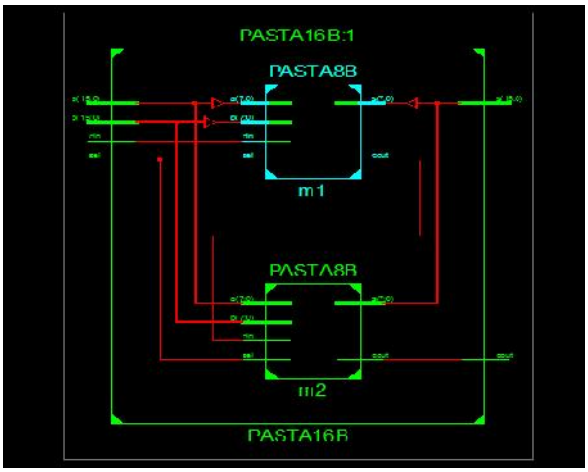


Figure 5: RTL schematic of internal block 16 bit PASTA adders

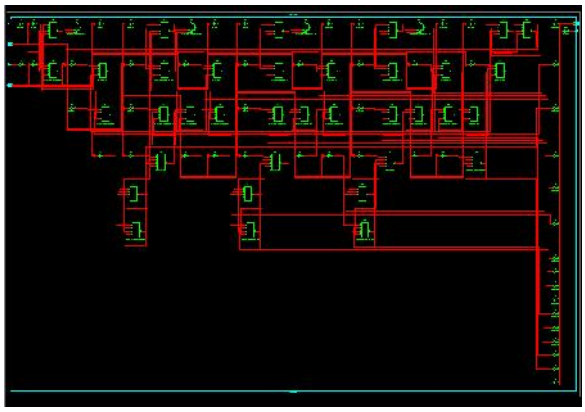


Figure 6: Technology schematic of Top-level 16 bit PASTA adders

PASTA16B Project Status			
Project File:	PASTA_ISE.xise	Parser Errors:	
Module Name:	PASTA16B	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	•Errors:	No Errors
Product Version:	ISE 14.4	•Warnings:	29 Warnings (29 new)
Design Goal:	Balanced	•Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	
Environment:	System Settings	•Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	20	4656	0%
Number of 4 input LUTs	36	9312	0%
Number of bonded IOBs	51	232	21%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed Aug 17 10:47:13 2016	0	29 Warnings (29 new)	0
Translation Report					
Map Report					
Place and Route Report					
Power Report					

Table 7-1: Design summary report of 16 bit PASTA adders

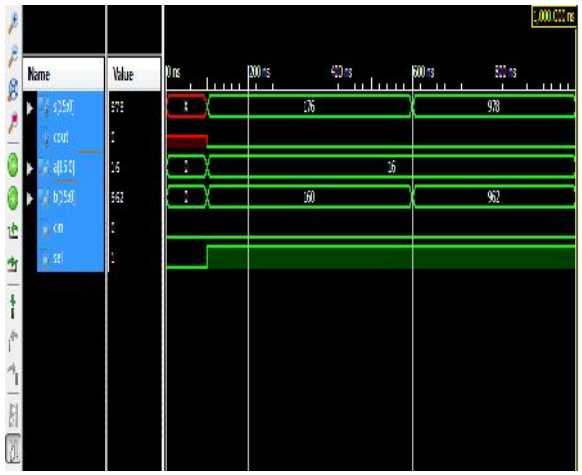


Figure 7: Simulated output for 16 bit PASTA adders

CONCLUSION AND FUTURE WORK

This brief presents an efficient implementation of PASTA. Initially, the theoretical foundation for a single-rail wave-pipelined adder is established. Subsequently, the architectural designs are presented. The design achieves a very simple n-bit adder that is area and interconnection-wise equivalent to the simplest adder namely the RCA. Moreover, the circuit works in a parallel manner for independent carry chains, and thus achieves logarithmic average time performance over random input values. The completion detection unit for the proposed adder is also practical and efficient. Simulation results are used to verify the advantages of the proposed approach.

REFERENCES

[1] D. Geer, "Is it time for clockless chips? [Asynchro-nous processorchips]," IEEE Comput., vol. 38, no. 3, pp. 18–19, Mar. 2005.

[2] J. Sparsø and S. Furber, Principles of Asynchronous Circuit Design.Boston, MA, USA: Kluwer Academic, 2001.

[3] P. Choudhury, S. Sahoo, and M. Chakraborty, "Im-plementation of basicarithmetic operations using cel-lular automaton," in Proc. ICIT, 2008,pp. 79–80.

[4] M. Z. Rahman and L. Kleeman, "A delay matched approach forthe design of asynchronous sequential circuits," Dept. Comput.Syst.Technol., Univ. Malaya, Kuala Lumpur, Malaysia, Tech. Rep. 05042013,2013.

[5] M. D. Riedel, "Cyclic combinational circuits," Ph.D. dissertation, Dept.Comput. Sci., California Inst. Tech-nol., Pasadena, CA, USA, May 2004.

[6] R. F. Tinder, Asynchronous Sequential Machine Design and Analysis:A Comprehensive Development of the Design and Analysis ofClock-Independent State Machines and Systems. San Mateo, CA, USA:Morgan, 2009.

[7] W. Liu, C. T. Gray, D. Fan, and W. J. Farlow, "A 250- MHz wavepipelined adder in 2-μm CMOS," IEEE J. Sol-id- State Circuits, vol. 29,no. 9, pp. 1117–1128, Sep. 1994.

[8] F.-C. Cheng, S. H. Unger, and M. Theobald, "Self-timed carry-lookaheadadders," IEEE Trans. Comput., vol. 49, no. 7, pp. 659–672, Jul. 2000.

[9] S. Nowick, "Design of a low-latency asynchronous adder using speculativecompletion," IEE Proc. Comput. Digital Tech., vol. 143, no. 5,pp. 301–307, Sep. 1996.

[10] N. Weste and D. Harris, CMOS VLSI Design: A Cir-cuits and SystemsPerspective. Reading, MA, USA: Ad-dison- Wesley, 2005.

[11] C. Cornelius, S. Koppe, and D. Timmermann, "Dy-namic circuit techniquesin deep submicron technolo-gies: Domino logic reconsidered," inProc. IEEE ICICDT, Feb. 2006, pp. 1–4. 2

[12] M. Anis, S. Member, M. Allam, and M. Elmasry, "Im-pact of technologyscaling on CMOS logic styles," IEEE Trans. Circuits Syst., AnalogDigital Signal Process., vol. 49, no. 8, pp. 577–588, Aug. 2002.